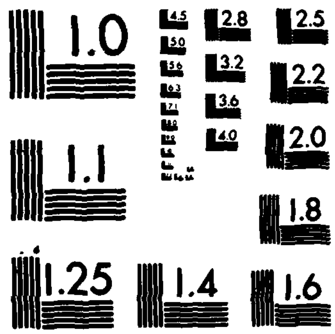
AD-A140 811 MINIMUM-ERROR TRELLIS-PATH DECODERS FOR CONVOLUTIONAL CODES(U) ADAPTIVE SENSORS INC SANTA MONICA CA I S REED MAR 84 N00019-83-C-0075 1/1

UNCLASSIFIED F/G 9/4 NL

END
FILMED
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

(12)

ASI Adaptive Sensors, Incorporated
216 Pico Blvd., Suite 8, Santa Monica, CA 90405 (213) 396 5997

MINIMUM-ERROR TRELLIS-PATH DECODERS FOR CONVOLUTIONAL CODES

I. S. Reed

March 1984

FINAL REPORT

N00019-83-C-0075

Submitted to

The Naval Air Systems Command
Washington, D.C. 20361

by

Adaptive Sensors, Incorporated
216 Pico Blvd, # 8
Santa Monica, CA 90405

CONTENTS

FIGURES

MINIMUM-ERROR TRELLIS-PATH DECODERS FOR CONVOLUTIONAL CODES

I. S. Reed

I. INTRODUCTION

This is the final report on a one-year study of techniques for improving and simplifying the decoding of convolutional codes (CCs). The approach to this problem was to first review and extend the present knowledge of the algebraic structure of CCs. This included a study [1] of both the generating and parity-check matrices and the relationships they satisfy. These ideas were then used to characterize the syndromes of an (n, k) CC and to find the most general solution of the syndrome equation.

Next, it was shown [2] how the concepts that were further developed in Ref. 1 could be used to realize a Viterbi-like syndrome decoding algorithm for (n, k) CCs. In order to develop a sequential syndrome decoder for CCs as well, the concept of an error tree or trellis was introduced [3]. Then, the Fano metric for use in sequential decoding was studied and subsequently modified [3] so that it now applies to sequential <u>syndrome</u> decoding of CCs.

In the last quarterly report [4], the concepts developed in the previous three reports were applied to the creation of a sequential syndrome decoding algorithm for the general (n, k) CC. For this new algorithm, it was shown how to find the syndrome of the received code, to sequentially solve the syndrome equation, to graph the solutions of the syndrome equation on an error tree or trellis, and, finally, how to utilize a modified form of Fano's metric to sequentially produce the minimum error sequence and to correct the received message. All parts of this new sequential decoding algorithm were illustrated in detail [4] by example.

In this final report of the study, a further conceptual simplification is made in the new syndrome decoding algorithm: the previous syndrome decoding technique is changed to avoid any explicit computation of the syndrome. This newest algorithm utilizes a modification of the solution of the syndrome equation developed earlier [1, 2, 4] to directly construct an error tree--or its more compact equivalent, a trellis. Within such a generated error trellis, the minimum weight path is found, either by Viterbi or other sequential decoding techniques. Finally, this best estimate of the path in the error trellis can be used to directly correct the noisy message produced from received code, z, by applying the right inverse G^{-1} of the generator matrix G to z.

Such a minimum-error trellis-path decoder for (n, k) CCs makes full and efficient use of the linear nature of convolutional codes. As such, it is expected that future physical realizations of syndrome decoders will be simpler and less costly than the present-day standard decoders for CCs.

II. DIRECT GENERATION OF ERROR-TRELLIS AND MINIMUM-ERROR DECODING

The fundamental algebraic structure of a CC, the generator and parity-check matrices associated with these codes, and how this information is used to compute the syndrome and to solve the syndrome equation for general (n, k) CCs are all presented in considerable detail in Refs. 1, 2, and 4. Here, only a brief synopsis of these concepts is given--just enough to systematically construct an error tree or trellis without resorting to the intermediary step of computing the syndrome.

The inputs and outputs of an (n, k) CC can be represented, respectively, as D-transforms

$$x(D) = \sum_{j=0}^{\infty} x_j D^j \tag{1}$$

and

$$y(D) = \sum_{i=0}^{\infty} y_j \, D^j \tag{2}$$

of the input sequence of k-vectors of form $x_j = \left[x_{1j}, x_{2j}, \ldots, x_{kj}\right]$ and the output sequence of n-vectors of form $y_j = \left[y_{1j}, y_{2j}, \ldots, y_{nj}\right]$, where x_{ij} and y_{ij} belong to a finite Galois field $F = G(q)$ usually restricted to the binary field GF(2) of two elements, and D is the delay operator. The input x(D) and the output y(D) are linearly related by means of a $k \times n$ generator matrix G(D) as follows:

$$y(D) = x(D)\; G(D) \ , \tag{3}$$

where the elements of G(D) are assumed usually to be polynomials over the finite field GF(q), where q is the power of a prime integer. The maximum degree M of the polynomial elements of G(D) is called the memory delay of the code, and the constant length of the code is $K = M + 1$.

In order to avoid catastrophic error propagation, the encoder matrix G(D) is assumed to be <u>basic</u> (see Refs. 2 and 4 for more discussion of this issue). For the basic encoder, the Smith normal form of G(D) is

$$G = A \left[I_k, \; 0\right] B \ , \tag{4}$$

where $A = A(D)$ is a $k \times k$ invertible matrix with elements in F[D], the ring of polynomials in D over F, and $B = B(D)$ is an $n \times n$ invertible matrix with elements in F[D]. The elements of the inverses A^{-1} and B^{-1} of matrices A and B, respectively, are polynomials in F[D].

By definition, the parity check matrix associated with $G = G(D)$ is any full-rank $(n - k) \times n$ matrix with polynomial elements in F[D] which satisfies

$$G(D)\; H^T\!\left(D\right) = 0 \ , \tag{5}$$

where "T" denotes matrix transpose. A modification of the method of Forney [5] is used to find H. The method involves a partitioning of matrix B in Eq. (4), as well as its inverse B^{-1}. That is, let

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{6}$$

and

$$B^{-1} = \left[\overline{B}_1, \overline{B}_2 \right] , \tag{7}$$

where the first k rows of B constitute submatrix B_1 and the remaining (n - k) rows are matrix B_2, and where, likewise, the first k columns of B^{-1} constitute submatrix $\overline{B}_1$ and the remaining (n - k) columns are matrix $\overline{B}_2$.

Since B times its inverse B^{-1} is the $n \times n$ identity matrix, the following identities hold (see Ref. 1):

$$\begin{aligned} B_1 \overline{B}_1 &= I_k , \quad B_1 \overline{B}_2 = 0 \\ B_2 \overline{B}_1 &= 0 \quad , \quad B_2 \overline{B}_2 = I_{n-k} . \end{aligned} \tag{8}$$

In terms of partition, Eq. (7), the Forney parity-check matrix is defined by

$$H = \overline{B}_2{}^T . \tag{9}$$

It is readily verified using Eq. (4) and identities of Eq. (8) that Eq. (9), in fact, satisfies Eq. (5), the requirement for H to be a parity-check matrix. It should be noted that the parity-check matrix is not unique. For example, it has been shown [1] that $H = C\, B_2{}^T$ is a parity-check matrix where C is any $(n - k) \times (n - k)$ invertible matrix with elements in F[D].

For an input message x(D) as defined in Eq. (1), the encoded message or code sequence is y(D) as generated by Eq. (3). Suppose y = y(D) is transmitted and that z = z(D) is received. Then, the transmitted and received

codes are related by

$$z(D) = y(D) + e(D) \ , \tag{10}$$

where e(D) is the D-transform of the error sequence. The <u>syndrome</u> of the received code z(D) is

$$s(D) = z(D) = H^T\Big(D\Big) \ . \tag{11}$$

If y(D) in Eq. (3) is substituted in Eq. (10), then the syndrome, computed in Eq. (11), satisfies, by Eq. (5),

$$\begin{aligned} s = z\ H^T &= \Big(x\ G + e\Big)\ H^T \\ &= e\ H^T \ . \end{aligned} \tag{12}$$

This is the syndrome equation for the error sequence e = e(D). The syndrome equation, Eq. (12) shows that the syndrome, computed by Eq. (11), is functionally independent of the original transmitted code y(D) as well as the original message x(D).

The problem of syndrome decoding of convolutional codes is, as for block codes, to solve the syndrome equation, Eq. (12), for the set of all possible solutions e = e(D). It has been shown [1] that this set of solutions is a coset of the set of all codewords.

To explicitly solve the syndrome equation, Eq. (12), substitute H as given by Eq. (9) in Eq. (12), thereby obtaining

$$s = e\ \bar{B}_2 = e\ B^{-1} \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} , \tag{13}$$

where I_{n-k} is the identity matrix of (n - k) rows. In Eq. (13), let

$$\varepsilon = e\ B^{-1} \ , \tag{14}$$

so that Eq. (13) becomes the simple equation

$$s = \varepsilon \begin{bmatrix} 0 \\ I_{n-k} \end{bmatrix} , \tag{15}$$

where $s = \left[s_1, s_2, \ldots, s_{n-k}\right]$ and $\varepsilon = \left[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k\right]$. The <u>general</u> solution of Eq. (15) over the ring F[D] is given evidently by

$$\begin{aligned} \left[\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k\right] &= \left[\tau_1, \tau_2, \ldots, \tau_k\right] \equiv \tau , \\ \left[\varepsilon_{k+1}, \varepsilon_{k+2}, \ldots, \varepsilon_n\right] &= \left[s_1, s_2, \ldots, s_{n-k}\right] = s , \end{aligned} \tag{16}$$

where $\tau_j = \tau_j\left(D\right)$ are <u>arbitrary</u> elements in F[D]. Thus, more compactly, the general solution of Eq. (14) is

$$\varepsilon = \left[\tau , s\right] = e\, B^{-1} , \tag{17}$$

where τ, as in Eq. (16), is an arbitrary k-vector of elements in ring F[D]. Finally, a multiplication of both sides of Eq. (17) by B yields

$$\begin{aligned} e = \varepsilon\, B &= \left[\tau , s\right] \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \\ &= \tau\, B_1 + s\, B_2 \end{aligned} \tag{18}$$

in terms of submatrices B_1 and B_2 in Eq. (6) as the most general solution of the syndrome equation, Eq. (12).

The general solution, Eq. (18), of the syndrome equation can be expressed in a number of different forms. For example, in Ref. 1, it is shown how it can be put into a canonical form originally found heuristically by Vinck, dePaepe, and Schalkwijk [6]. Herein, it is desired to put Eq. (18) in a form which makes it possible in the syndrome decoding process to bypass the explicit computation of syndrome s(D).

Towards this end, note first from Eq. (4), the Smith normal form of the basic encoder, that

$$A^{-1} G = \left[I_k, 0\right] B = B_1 .$$

Next, substitute into Eq. (18) both this value for B_1 and, by Eqs. (9) and (11), the quantity $z\,\overline{B}_2$ for syndrome s. These substitutions yield

$$e = \left(\tau\ A^{-1}\right) G + z\left(\overline{B}_2\ B_2\right) \tag{19}$$

in terms of received code z for the general solution of the syndrome equation.

Since τ is an arbitrary k-vector,

$$t = \tau\ A^{-1} \tag{20}$$

is also an arbitrary k-vector of elements in F[D]. In Eq. (19), let R be the $n \times n$ matrix $\overline{B}_2\ B_2$. Since B_2 and $\overline{B}_2$ have ranks $(n - k)$, it can be shown that matrix $R = \overline{B}_2\ B_2$, where B_2 and $\overline{B}_2$ are defined in Eqs. (6) and (7), respectively, also has rank $(n - k)$.

A substitution of Eq. (20) and R into Eq. (19) yields

$$e = t\,G + z\,R \tag{21}$$

as the general solution of the syndrome equation. Here, R is the $n \times n$, rank $(n - k)$ matrix

$$R = \overline{B}_2\ B_2 , \tag{22}$$

t is an arbitrary k-vector of elements in F[D], and z is the D-transform of the received code sequence.

Let z(D) be any finite-length received code. By the maximum likelihood principle, the most likely error sequence is the one with minimum Hamming weight. Given z(D), the sequence e(D) with minimum Hamming weight is found

by minimizing the weight of the right side of Eq. (21) over all polynomials $t(D)$ in $F[D]$. That is,

$$\text{Min} \left\| e \right\| = \underset{t \,\varepsilon\, F[D]}{\text{Min}} \left\| t\, G + z\, R \right\| , \tag{23}$$

where $z = z(D)$ is the D-transform or polynomial of any finite-length received code and "$\| x \|$" denotes the Hamming weight or "norm" of element $x = x(D)$ in $F[D]$.

The minimization required in Eq. (23) is analogous to certain optimum nulling techniques in control. Sequence $r(D) = z(D)\, R(D)$ is the error sequence for the zero input message, i.e., for $t(D) = 0$. What one attempts to do in Eq. (23) is to find that message sequence $\hat{t}$ which, when encoded as $\hat{t}\, G$ and subtracted from $r(D)$, yields the sequence $\hat{e}$ of minimum Hamming weight. That is, if $\hat{t} = \hat{t}(D)$ is the D-transform for which $\| e \| = \| t\, G + z\, R \|$ is a minimum, then

$$\hat{e} = \hat{t}\, G + z\, R \tag{24}$$

is the D-transform of the minimum-weight-possible error sequence.

By Eq. (4), the right inverse G^{-1} of the generating matrix G is

$$G^{-1} = B^{-1} \begin{bmatrix} I_k \\ 0 \end{bmatrix} A^{-1} . \tag{25}$$

This is verified by multiplying G in Eq. (4) on the right by G^{-1} in Eq. (25). Multiplying both sides of Eq. (24) on the right by G^{-1} in Eq. (25) yields, by Eqs. (7) and (8), the identity

$$\begin{aligned} \hat{e}\, G^{-1} &= \left[\hat{t}\, G + z\, \bar{B}_2\, B_2 \right] G^{-1} = \hat{t} + z\, \bar{B}_2\, B_2 \left[\bar{B}_1 , \bar{B}_2 \right] \begin{bmatrix} I_k \\ 0 \end{bmatrix} A^{-1} \\ &= \hat{t} + z\, \bar{B}_2 \left[0 , I_{n-k} \right] \begin{bmatrix} I_k \\ 0 \end{bmatrix} A^{-1} = \hat{t} . \end{aligned} \tag{26}$$

By Eq. (10), the subtraction of $\hat{e}$ from z produces a best estimate $\hat{y}$ of the transmitted code, i.e.,

$$\hat{y} = z + \hat{e} . \tag{27}$$

The best estimate $\hat{y}$ of the code, if multiplied on the right by G, yields

$$\hat{x} = \hat{y}\, G^{-1} , \tag{28}$$

the best estimate of the original message. Hence, substituting Eq. (27) in Eq. (28) and using Eq. (26) produces

$$\begin{aligned} \hat{x} &= \left(z + \hat{e}\right) G^{-1} \\ &= z\, G^{-1} + \hat{t} . \end{aligned} \tag{29}$$

This important identity shows that $\hat{t} = \hat{t}(D)$, obtained by the minimization in Eq. (23), is a correction factor to the standard method of recovering the message from $z = z(D)$ if z were noise-free.

In the following section, the techniques of performing the minimization in Eq. (23) for finding $\hat{e}$ and $\hat{t}$ are discussed. Included in these methods are the Viterbi dynamic programming algorithm and all of the sequential decoding techniques. An interesting new suboptimal method for finding $\hat{e}$ is the piecewise L-step minimum-error path technique for finding $\hat{e}$, developed below.

III. PIECEWISE L-STEP MINIMUM-ERROR DECODING

In this section, a new piecewise L-step minimum-error decoder is developed by example. The encoder is the (3, 1) CC described in Ref. 6, Chapter 12, and also used in Ref. 4 for an example. For ease of description and for comparative purposes, many of the parameters used in Refs. 4 and 6 are also used here.

The generator matrix of the above-mentioned (3, 1) CC is

$$G = \left[1 + D,\ 1 + D^2,\ 1 + D + D^2\right] . \tag{30}$$

The Smith normal form, Eq. (4), of this generator is

$$G = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} B \,, \tag{31}$$

where

$$B = \begin{bmatrix} 1 + D, & 1 + D^2, & 1 + D + D^2 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} . \tag{32}$$

The inverse of B is obtained readily as

$$B^{-1} = \begin{bmatrix} 1, & 1 + D^2, & D \\ 1, & D^2 & , 1 + D \\ 1, & 1 + D^2, & 1 + D \end{bmatrix} . \tag{33}$$

Applying Eqs. (6) and (7) to Eqs. (32) and (33) yields

$$B_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \overline{B}_2 = \begin{bmatrix} 1 + D^2, & D \\ D^2 & , 1 + D \\ 1 + D^2, & 1 + D \end{bmatrix} . \tag{34}$$

Hence, by Eq. (22), matrix R is

$$R = \overline{B}_2 \, B_2 = \begin{bmatrix} D & , 1 + D^2, & 1 + D + D^2 \\ 1 + D, & D^2 & , 1 + D + D^2 \\ 1 + D, & 1 + D^2, & D + D^2 \end{bmatrix} , \tag{35}$$

and, in Eq. (21),

$$\begin{aligned} r = z\,R &= \begin{bmatrix} z_1, & z_2, & z_3 \end{bmatrix} R \\ &= \Big[D\,z_1 + \left(1 + D\right) z_2 + \left(1 + D\right) z_3, \left(1 + D^2\right) z_1 + D^2\, z_2 + \left(1 + D^2\right) z_3, \\ &\qquad \left(1 + D + D^2\right) z_1 + \left(1 + D + D^2\right) z_2 + \left(D + D^2\right) z_3 \Big] . \end{aligned} \tag{36}$$

Use the data in the example of Ref. 4 for z; namely,

$$z_1 = \left[1\ 1\ 1\ 1\ 0\ 1\ 0\right],$$

$$z_2 = \left[1\ 1\ 1\ 1\ 1\ 0\ 0\right],$$

and

$$z_3 = \left[0\ 0\ 0\ 1\ 1\ 1\ 1\right].$$

Then, the r-sequence z R in Eq. (36), needed for Eq. (23), is computed as follows:

$$\begin{array}{r|ccccccccc}
z_2: & 1 & 1 & 1 & 1 & 1 & 0 & 0 & & \\
D\,z_2: & & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \\
D\,z_1: & & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
z_3: & 0 & 0 & 0 & 1 & 1 & 1 & 1 & & \\
D\,z_3: & & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
\hline
r_1 = & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0
\end{array}$$

$$\begin{array}{r|ccccccccc}
z_1: & 1 & 1 & 1 & 1 & 0 & 1 & 0 & & \\
D^2\,z_1: & & & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
D^2\,z_2: & & & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
z_3: & 0 & 0 & 0 & 1 & 1 & 1 & 1 & & \\
D^2\,z_3: & & & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
r_2 = & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1
\end{array}$$

$$\begin{array}{r|ccccccccc}
z_1: & 1 & 1 & 1 & 1 & 0 & 1 & 0 & & \\
D\,z_1: & & 1 & 1 & 1 & 1 & 0 & 1 & 0 & \\
D^2\,z_1: & & & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
z_2: & 1 & 1 & 1 & 1 & 1 & 0 & 0 & & \\
D\,z_2: & & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \\
D^2\,z_2: & & & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
D\,z_3: & & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
D^2\,z_3: & & & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
r_3 = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}$$

Hence, the sequence r as a D-transform is

$$r(D) = z(D)\,R(D) = \begin{bmatrix}1 & 1 & 0\end{bmatrix} + \begin{bmatrix}1 & 1 & 0\end{bmatrix} D + \begin{bmatrix}1 & 1 & 0\end{bmatrix} D^2$$

$$+ \begin{bmatrix}1 & 1 & 0\end{bmatrix} D^4 + \begin{bmatrix}1 & 1 & 0\end{bmatrix} D^5$$

$$+ \begin{bmatrix}1 & 1 & 0\end{bmatrix} D^6 + \begin{bmatrix}1 & 0 & 1\end{bmatrix} D^7 + \begin{bmatrix}0 & 1 & 1\end{bmatrix}$$

or simply as a sequence

$$r = \begin{bmatrix}1\,1\,0, & 1\,1\,0, & 1\,1\,0, & 0\,0\,0, & 1\,1\,0\,, & \underline{1}\,1\,0, & 1\,1\,0, & 1\,0\,1, & 0\,1\,1\end{bmatrix}. \qquad (37)$$

The trellis diagram for the encoder of the (3, 1) CC is shown in Fig. 1. Next, in Fig. 2, is shown the one-step minimum-error path for sequence r in Eq. (37), the example in Ref. 4. The sequence r = z R is shown at the top of the partial trellis as a function of step. r would have been the error sequence if the message had been x = 0. At steps 4 and 7, for example, $r = \begin{bmatrix}1 & 1 & 0\end{bmatrix}$ and $r = \begin{bmatrix}1 & 0 & 1\end{bmatrix}$, respectively.

The path in Fig. 2 is obtained by performing the minimization in Eq. (23) on a step-by-step basis. For example, at step 0 by the encoder trellis in Fig. 1, there are two possible paths: one path has, by Eq. (21), e = 1 1 0 - 0 0 0 = 1 1 0, and the other has e = 1 1 0 - 1 1 1 = 0 0 1. Since the latter path has minimum Hamming weight, the path a $\rightarrow$ c is chosen with $\hat{t} = 1$. Next, at step 1 and starting at state c, there are again two possible choices: by Fig. 3, one choice is path c $\rightarrow$ b with e = 1 1 0 - 1 0 1 = 0 1 1; the other choice is path c $\rightarrow$ d with e = 1 1 0 - 0 1 0 = 1 0 0. Again, since $\|0\;1\;1\| = 2 > \|1\;0\;0\| = 1$, path c $\rightarrow$ d is chosen and $\hat{t} = 1\;1$. This process repeats until, finally, $\hat{t} = \begin{bmatrix}1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0\end{bmatrix}$ up until step 9.

It is evident that the simple one-step minimum-error path decoding process illustrated above produces a value of $\hat{t}$ which is the same as the value of $\hat{t}$ found by the substantially more complicated sequential decoding processes described in both Ref. 4 and Ref. 6 for moderate noise levels. In fact, this

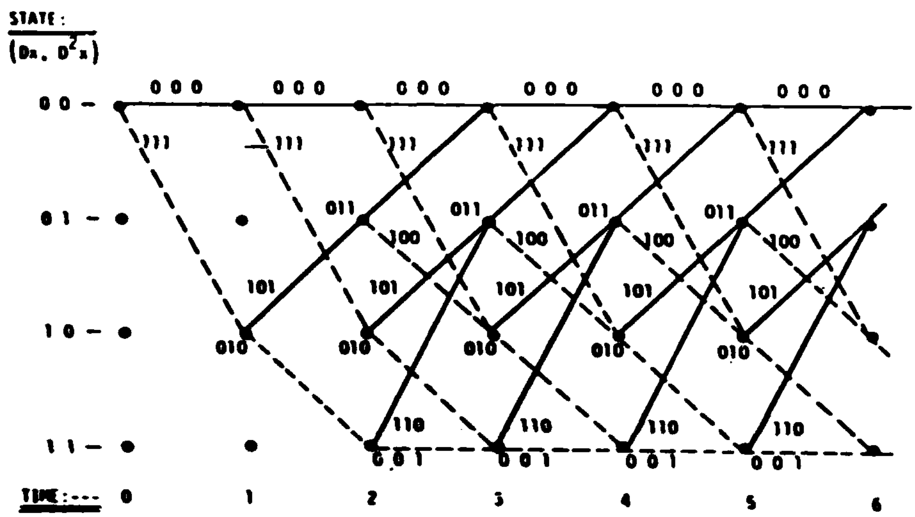


Fig. 1 — Trellis diagram for encoder of (3, 1) convolutional code

[Note: Dashed line corresponds to x = 1;
bold line corresponds to x = 0]

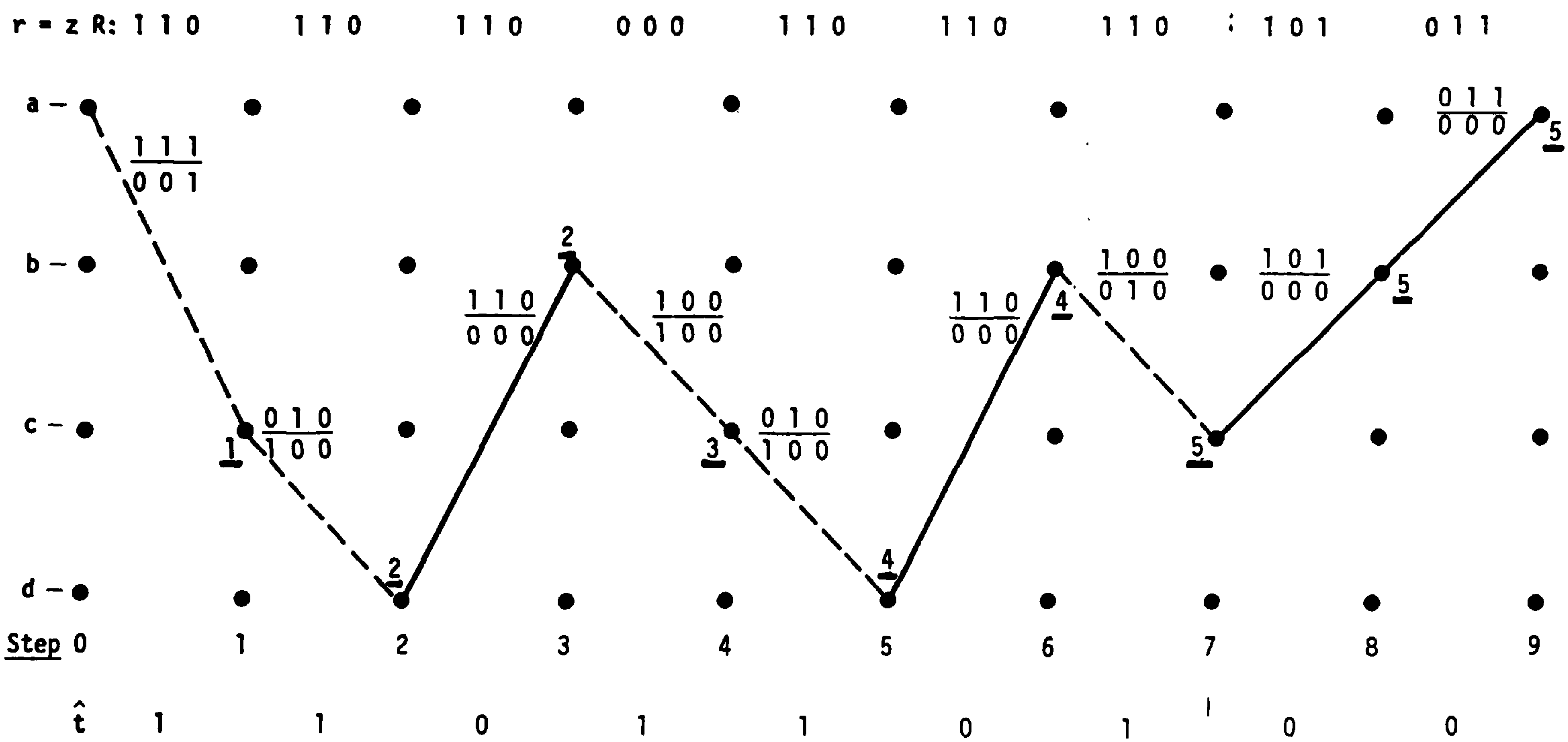


Fig. 2 — One-step minimum-error path decoding for example given in Ref. 4

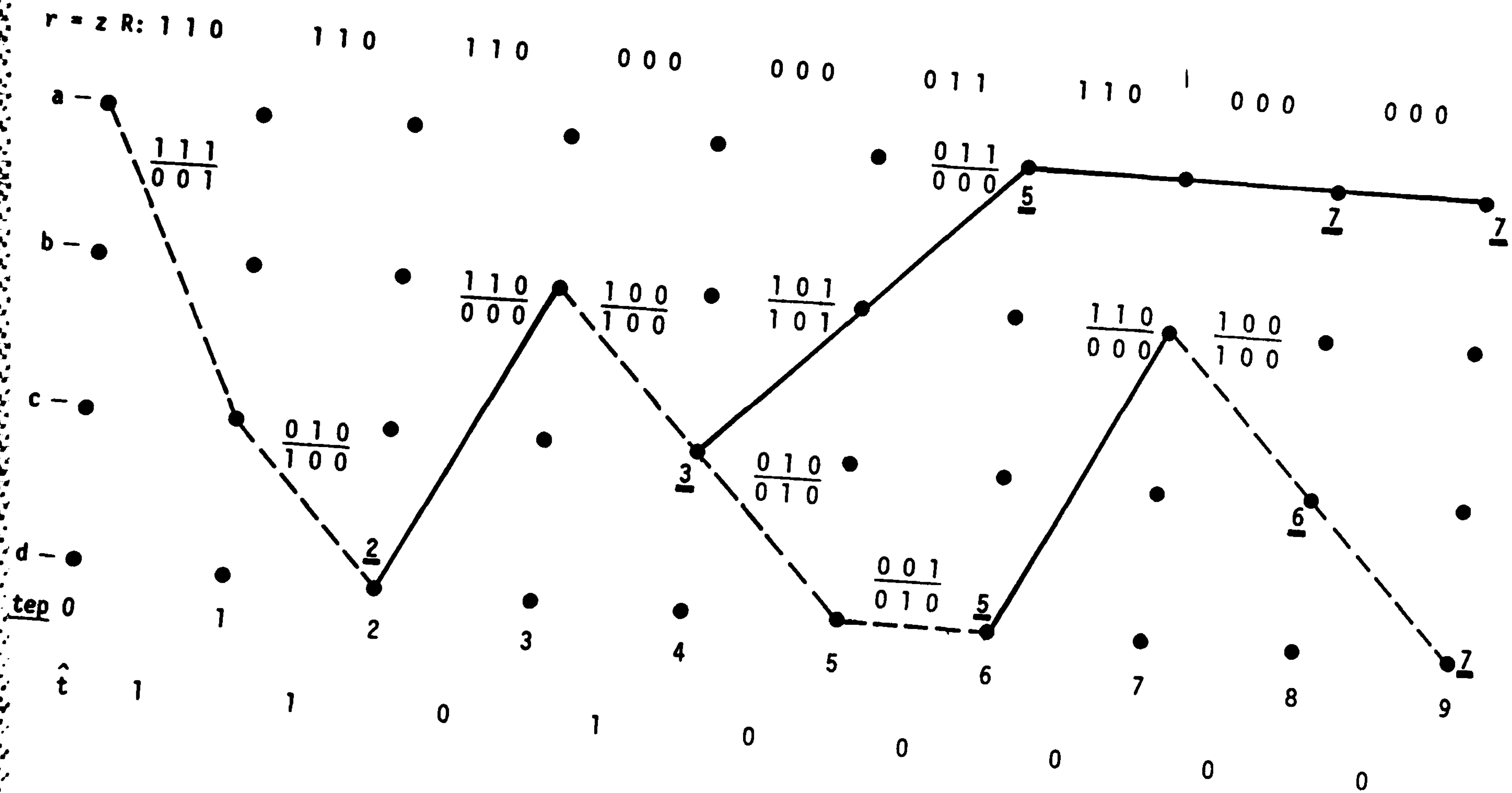


Fig. 3 — Piecewise two-step minimum-error path decoding for very noisy example given in Ref. 6, p. 357

new decoding process is ideally suited to high-rate (n, k) convolutional codes, where the channel noise is moderate.

The piecewise two-step minimum-error decoding procedure is illustrated for the "very noisy" example in Ref. 5 (p. 357, Example 12.5). For this example,

$$z = \left[1\,1\,0,\ 1\,1\,0,\ 1\,1\,0,\ 1\,1\,1,\ 0\,1\,0,\ 1\,0\,1,\ 1\,0\,1 \right] \tag{38}$$

and $r = z\,R$ is computed, by Eq. (36), to be

$$r = \left[1\,1\,0,\ 1\,1\,0,\ 1\,1\,0,\ 0\,0\,0,\ 0\,0\,0,\ 0\,1\,1,\ 1\,1\,0,\ 0\,0\,0,\ 0\,0\,0 \right],$$

which is listed at the top of the partial trellis in Fig. 3. In the two-step process, minimum path segments of two steps are found and pieced together to form an estimate of $\hat{t}$.

Starting at step 0 and state a, the values of e are found using Eq. (21) and Fig. 1 for the four path segments between steps 0 and 2. The Hamming weights of the four possible two-step segments are used then to determine the two-step path of minimum weight, which, in this case, is the unique path $a \rightarrow c \rightarrow d$ of weight 2 and with $\hat{t} = 1\ 1$. In a similar fashion, the unique minimum two-step path segment, starting at step 2 at state d, is $d \rightarrow b \rightarrow c$. As a consequence, the minimum path between step 0 and step 4 is $a \rightarrow c \rightarrow d \rightarrow b \rightarrow c$, for which $\hat{t} = 1\ 1\ 0\ 1$.

The two-step segment of path between steps 4 and 6 is not unique. A minimization over the four possible paths yields two path segments, $c \rightarrow b \rightarrow a$ and $c \rightarrow d \rightarrow d$, both of which equal Hamming weight 2. From one point of view, such a tie could be called a decoding failure, particularly since the free distance, d_{free}, of 3 for this code has been exceeded along either path by a total Hamming distance of 5 within a code length which about equals the minimum truncation length, τ_{min}, of $5 \cdot 3 = 15$. However, it is more customary with convolutional

codes to continue to track both paths until either they merge or one track definitely has a smaller Hamming weight than the other. If such a criterion is incorporated into the decoding algorithm, the upper path, namely, a → c d → b → c → b → a → a → a → a, with

$$\hat{t} = \left[\, 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0 \,\right], \tag{39}$$

eventually wins out.

Given that the received sequence z is Eq. (38) and that the decoder produces $\hat{t}$ as the estimated correction factor, it is now desired to find $\hat{x}$, the estimated message from Eq. (29). By Eqs. (25), (31), and (33),

$$G^{-1} = B^{-1}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1, & 1 + D^2, & D \\ 1, & D^2 & , 1 + D \\ 1, & 1 + D^2, & 1 + D \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Hence, by Eq. (29),

$$\hat{x} = z\, G^{-1} + \hat{t} = \left[z_1,\ z_2,\ z_3\right]\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \hat{t}$$

$$= z_1 + z_2 + z_3 + \hat{t}\ . \tag{40}$$

Applying Eq. (40) to z and $\hat{t}$ in Eqs. (38) and (39), respectively, yields

$$\begin{array}{ll} z_1 : & 1\ 1\ 1\ 1\ 0\ 1\ 1 \\ z_2 : & 1\ 1\ 1\ 1\ 1\ 0\ 0 \\ z_3 : & 0\ 0\ 0\ 1\ 0\ 1\ 1 \\ \hat{t}\ : & 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ \hline \hat{x}\ = & \left[\, 1\ 1\ 0\ 0\ 1\ 0\ 0 \,\right]. \end{array}$$

Hence,

$$\hat{x}(D) = 1 + D + D^4$$

is the best estimate of the transmitted message. This agrees with the result obtained by Lin and Costello [6, p. 358] using the stack sequential decoding algorithm.

The piecewise L-step minimum-error decoding partially developed in this section to find the minimum-error trellis path by the criterion of Eq. (23) is sufficiently simple compared with other decoding critteria to warrant more study. For example, on heuristic grounds and to achieve the best results, it is believed that L should either equal M, the memory, or K = M + 1, the constraint length of the code.

IV. CONCLUDING REMARKS

In this final report on a one-year study of syndrome decoding techniques, the results of previous studies [1, 2, 3, 4] are extended and applied to the development of a new class of minimum-error trellis-path decoders for (n, k) convolutional codes. Also, a new method is devised by example for finding the minimum-error trellis path. This promising technique is called piecewise L-step minimum-error decoding. Other more standard decoding methods can also be used to find the minimum-error trellis path, such as a Viterbi-like algorithm and the various sequential decoding techniques.

It is difficult to estimate the importance to the military of improving the capability of convolutional decoders. Right now, there is no known method for satisfactorily decoding high-rate convolutional codes and, in most communications, high-rate transmission is very desirable. Viterbi decoders cannot be designed with the present VHSIC technology to handle convolutional codes of constraint lengths of greater than 7 and with rates greater than 1/2. Finally,

sequential decoders work fine in a benign environment. But, with even the slightest interference or jamming, sequential decoding exhibits long queues or almost complete decoding failure.

In view of these reasons, much attention has been given in the present study to new algorithms which might lead to substantial reductions in cost and complexity while achieving decoding realiability. It is believed that the class of minimum-error trellis-path decoders developed in this and the previous reports submitted on this study provide a new and feasible approach to these important goals.

REFERENCES

1. Reed, I. S., *Canonical Solutions of the Syndrome Equation for (n, k) Convolutional Codes*, First Quarterly Report submitted to the Naval Air Systems Command on Contract N00019-83-C-0075 by Adaptive Sensors, Inc., August 1983.

2. Reed, I. S., *New Syndrome Decoding Techniques for Convolutional Codes Over GF(q)*, Final Report submitted to Naval Air Systems Command on Contract N00019-81-C-0541 by Adaptive Sensors, Inc., January 1983.

3. Reed, I. S., *A Modification of the Fano Metric for Sequential Syndrome Decoding of Convolutional Codes*, Second Quarterly Report submitted to the Naval Air Systems Command on Contract N00019-83-0075 by Adaptive Sensors, Inc., October 1983.

4. Reed, I. S., *Sequential Syndrome Decoding of Convolutional Codes*, Third Quarterly Report submitted to the Naval Air Systems Command on Contract N00019-83-C-0075 by Adaptive Sensors, Inc., January 1984.

5. Forney, G. D., "Convolutional Codes: Algebraic Structures," *IEEE Transactions on Information Theory*, IT-6, 1970, pp. 720-738.

6. Lin, S., and D. J. Costello, Jr., *Error Control Coding*, Princeton, New Jersey, 1983.

END

FILMED

6-84

DTIC